

VISUALISASI KETERHUBUNGAN ANTARISTILAH SUBJEK PADA METADATA TESIS IPB MENGGUNAKAN PELUANG BERSYARAT

Reza Risky¹, Firman Ardiansyah²

¹Mahasiswa Departemen Ilmu Komputer, Fakultas Matematika dan IPA,
Institut Pertanian Bogor

²Dosen Departemen Ilmu Komputer, Fakultas Matematika dan IPA,
Institut Pertanian Bogor, email: f.ardiansyah@ipb.ac.id

Abstract

Bogor Agricultural University library has metadata for its collection. The metadata is used for searching and browsing. The activity of searching and browsing can be based on metadata fields, such as title, author, subject, etc. Searching on the subject or topic is an effective way if someone want to search document that fulfill his/her topic. When someone search on a subject, the result is only documents that has the subject. So, someone needs to pick an appropriate query to find the desirable result. The decision for picking a query is based on how good the knowledge of someone about the collection. It will be hard if someone does not know about the collection. In that case, we need an application that shows the relationship between the subject, so the related terms can be used as query suggestion for someone to decide his/her query. In order to make the view of relationship more interesting, understandable, and interactive, the relationship is visualized. The result of this research is an implementation of a Java desktop application that searches document based on subject and visualize subject terms relationship. The documents are from IPB thesis metadata. From 7445 documents, 7416 documents has subject field. Obtained 4496 unique terms from the 7416 document. From test result, the process to show document result and visualization is fast enough. The average time to show document result and visualization is one second.

Keywords: *visualization, conditional probability, terms relationship.*

1. Pendahuluan

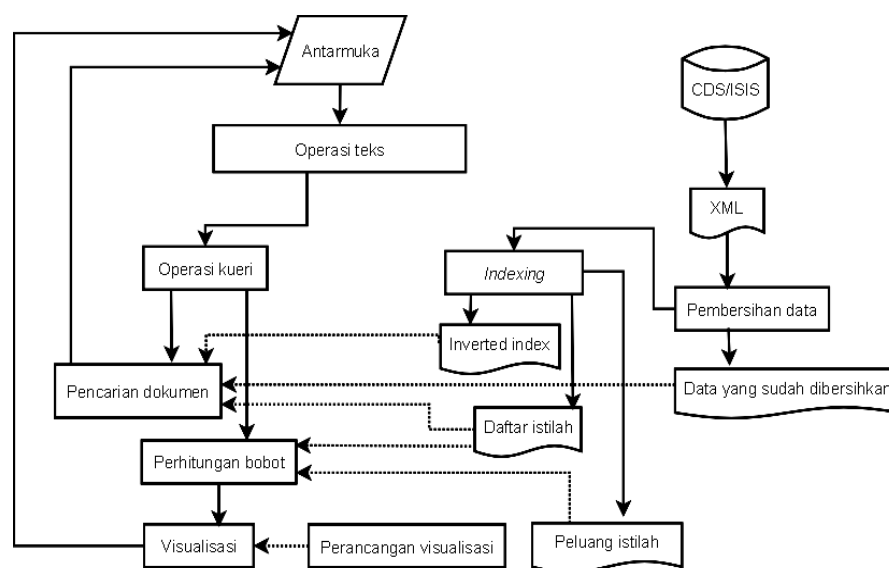
Perpustakaan pusat IPB memiliki metadata pada koleksi yang dimilikinya. Metadata tersebut biasa digunakan sebagai sarana pencarian atau penelusuran. Pencarian dan penelusuran koleksi bisa berdasarkan *field* yang terdapat pada metadata, yaitu berdasarkan judul, pengarang, subjek dan lain-lain.

Pencarian dan penelusuran suatu koleksi merupakan suatu kegiatan yang sering terjadi di perpustakaan. Suatu kegiatan pencarian dilakukan jika seseorang sudah tahu apa yang akan dicarinya sedangkan penelusuran dilakukan jika seseorang belum tahu pasti apa yang dicarinya sampai dia menemukan pilihan yang sesuai dengan yang diinginkannya.

Pencarian berdasarkan subjek atau topik merupakan cara yang efektif jika

seseorang ingin mencari dokumen yang sesuai dengan topik yang diinginkan. Ketika seseorang memasukkan suatu kueri untuk mencari berdasarkan suatu subjek, maka hanya akan dihasilkan isi koleksi yang memiliki subjek tersebut. Untuk mendapatkan hasil yang sesuai diperlukan pemilihan kueri yang tepat. Penentuan kueri yang tepat tersebut tergantung dari seberapa baik orang tersebut dalam mengetahui subjek-subjek yang terdapat dalam dokumen. Hal ini tentu akan sulit dilakukan bagi pihak yang tidak begitu tahu isi dokumen yang ada pada suatu koleksi.

Oleh karena itu, dibutuhkan suatu aplikasi yang dapat menunjukkan keterhubungan antaristilah pada subjek. Dengan ditampilkannya keterhubungan tersebut maka istilah-istilah yang memiliki nilai keterhubungan dengan kueri bisa



Gambar 1 Metode Penelitian.

digunakan sebagai saran sehingga seseorang bisa mempunyai pilihan kueri yang bisa ditelusuri.

Untuk membuat keterhubungan antaristilah lebih menarik, lebih mudah dimengerti dan lebih interaktif maka keterhubungan antaristilah tersebut akan divisualisasikan.

Tujuan penelitian ini adalah untuk mendapatkan keterhubungan istilah pada subjek dengan menggunakan peluang bersyarat dan memvisualisasikannya.

Ruang lingkup penelitian ini adalah :

1. Data yang digunakan data tesis perpustakaan pusat IPB tahun 2006 yang menggunakan format CDS/ISIS yang sudah diekspor ke dalam format XML.
2. Data XML yang digunakan harus *well-formed*.
3. Pembuatan *index* istilah subjek berdasarkan kata bukan frase.

Penelitian ini diharapkan dapat membantu seseorang ketika menelusuri isi dari suatu koleksi, bukan hanya menampilkan apa yang dia cari tapi juga memberikan saran kueri yang digunakan untuk mengembangkan pencarian yang dilakukannya.

2. Metode Penelitian

Penelitian ini dapat dibagi dalam beberapa tahap. Secara garis besar, tahapannya adalah pengumpulan data, pembersihan data, *indexing*, perancangan visualisasi, pencarian dokumen, penghitungan bobot, dan antarmuka hasil hasil pencarian dan penghitungan bobot. Untuk lebih jelas, dapat dilihat ilustrasi metode penelitian ini pada Gambar 1.

2.1. Pengumpulan Data

Pada penelitian ini digunakan metadata tesis perpustakaan IPB pusat tahun 2006 yang menggunakan CDS/ISIS. Metadata tersebut mengikuti format INDOMARC dalam penentuan *field* yang digunakan.

2.2. Pembersihan Data

Metadata dalam bentuk CDS/ISIS diekspor ke dalam bentuk XML. XML yang dihasilkan harus *well-formed*. XML yang tidak memenuhi kriteria maka akan diperbaiki.

2.3. Indexing

Proses *indexing* meliputi dua proses, yaitu pembuatan peluang istilah dan pembuatan *inverted index* serta daftar istilah. Untuk proses *indexing* ini, dilakukan proses parsing pada XML hasil ekspor CDS/ISIS.

2.4. Penghitungan Peluang Istilah dengan Rumus

Dalam *indexing*, dilakukan pencarian istilah unik lalu membuat daftar istilah unik. Setelah membuat daftar istilah, dicari peluang istilah dengan menghitung peluang bersyarat setiap istilah dengan menggunakan rumus :

$$P(t_i | t_j) = \frac{\text{jumlah dokumen yang ada } t_i \text{ dan } t_j \dots (1)}{\text{jumlah dokumen yang ada } t_j}$$

$P(t_i | t_j)$ adalah peluang munculnya istilah t_i setelah kemunculan t_j . Jika terdapat 100 istilah unik maka akan menghasilkan matriks berukuran 100 x 100. Matriks yang dihasilkan bukan matriks simetris karena hasil antara $P(t_i | t_j)$ berbeda dengan $P(t_j | t_i)$.

Misalkan didapat empat istilah unik, maka nilai yang terbentuk ditunjukkan pada Tabel 1.

Tabel 1 Contoh penghitungan peluang

	t_1	t_2	t_3
t_1	$P(t_1 t_1)$	$P(t_1 t_2)$	$P(t_1 t_3)$
t_2	$P(t_2 t_1)$	$P(t_2 t_2)$	$P(t_2 t_3)$
t_3	$P(t_3 t_1)$	$P(t_3 t_2)$	$P(t_3 t_3)$

2.5. Pembuatan *Inverted Index*

Proses *indexing* berikutnya adalah pembuatan *inverted index* untuk digunakan pada pencarian dokumen. Pada proses pembuatan *inverted index* ini ditulis istilah unik beserta dokumen yang bersesuaian dengan istilah tersebut.

2.6. Perancangan Visualisasi

Perancangan visualisasi dilakukan untuk membuat visualisasi yang akan digunakan setelah mendapat hasil penghitungan bobot istilah. Perancangan visualisasi ini dibagi menjadi tiga tahap, yaitu pemilihan bentuk visual dasar, perbaikan bentuk visual dasar, dan penambahan fitur interaksi.

2.7. Pemilihan Bentuk Visual Dasar

Pemilihan bentuk visual dasar yang dapat digunakan untuk memvisualisasikan

informasi tersebut. Misal informasi tersebut divisualisasikan dalam bentuk *bar*, *pie chart*, *list*, *graph* atau *tree*.

2.8. Perbaikan Bentuk Visual

Bentuk visual yang didapat dari tahap sebelumnya diperbaiki sehingga representasi yang ditampilkan lebih jelas dan tampilannya lebih menarik.

2.9. Penambahan Fitur Interaksi

Ditambahkan fitur-fitur interaksi pada visualisasi yang sudah ada agar penggunaannya lebih memudahkan pengguna.

2.10. Pencarian dokumen

Ketika pengguna memasukkan input kueri berupa subjek yang ingin dicari, maka akan dilakukan operasi teks pada input kueri untuk memisahkan kata atau *term*, lalu diproses kata-kata tersebut dengan operasi kueri. Setelah itu dilakukan pencarian dokumen dan penghitungan bobot istilah. Proses pencarian dokumen akan mencari dokumen dengan subjek yang sesuai dengan kueri.

2.11. Penghitungan Bobot Istilah

Penghitungan bobot istilah digunakan untuk mencari istilah-istilah yang berdekatan dengan kueri. Untuk istilah yang berada di kueri dan di daftar istilah diberi nilai 1, sedangkan untuk istilah lain yang tidak ada di dalam kueri, dihitung bobot antara istilah tersebut dengan kueri menggunakan rumus :

$$r_i = \frac{P(t_i | t_1) + P(t_i | t_2) \dots (2)}{x}$$

Dengan x adalah jumlah istilah yang memiliki bobot istilah bernilai satu.

Setelah itu nilai tersebut diurutkan dari nilai yang terbesar. Semakin besar nilai maka semakin dekat hubungan antaristilah, istilah yang berasal dari kueri tidak perlu dihitung lagi. Contoh jika pada daftar istilah terdapat 4 istilah, yaitu t_1 , t_2 , t_3 , dan t_4 lalu diberikan kueri " $t_1 t_2$ " maka penghitungan dilakukan seperti pada Tabel 2.

Tabel 2 Contoh penghitungan bobot istilah

	Bobot Istilah
t_1	1
t_2	1
t_3	$(P(t_3 t_1) + P(t_3 t_2)) / 2$
t_4	$(P(t_4 t_1) + P(t_4 t_2)) / 2$

2.12. Antarmuka

Hasil dari pencarian dokumen ditampilkan ke antarmuka aplikasi. Untuk hasil penghitungan bobot istilah, divisualisasikan terlebih dahulu mengikuti hasil visualisasi dari perancangan visualisasi sebelumnya lalu ditampilkan ke antarmuka aplikasi.

2.13. Lingkungan Pengembangan Sistem

Spesifikasi beberapa perangkat lunak dan perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Perangkat Lunak :

- Sistem operasi Microsoft Windows XP
- Java Development Kit 1.6 untuk pengembangan aplikasi berbasis Java
- Prefuse visualization framework untuk pembuatan visualisasi
- SwingX untuk komponen tambahan Swing
- Netbeans IDE 6.7

2. Perangkat Keras :

- Intel Pentium 4 3.0 Ghz
- Memory 1536 MB RAM
- Monitor resolusi 1024 x 768 *pixel*
- *Mouse* dan *keyboard*

3. Hasil dan Pembahasan

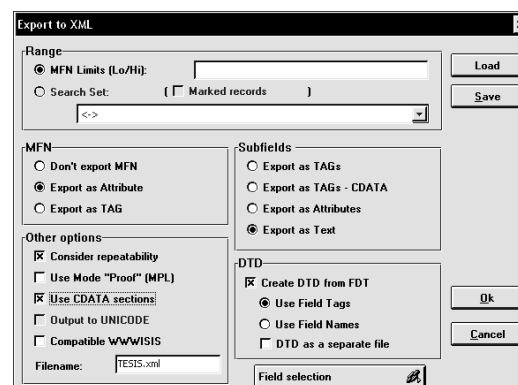
3.1. Pengumpulan Data

Penelitian ini menggunakan 7445 dokumen yang berisi metadata tesis perpustakaan pusat IPB tahun 2006 yang

menggunakan CDS/ISIS. Nama-nama *field* yang digunakan pada metadata mengikuti format INDOMARC di mana setiap *field* memiliki nomor *tag* tersendiri. Contoh *field* yang terdapat pada metadata tesis adalah :

- No Registrasi (*tag* 999)
- Pengarang utama (*tag* 985)
- Judul (*tag* 245)
- No. UDC (*tag* 80)
- Subjek (*tag* 650)

Data dalam CDS/ISIS tersebut diekspor ke dalam bentuk XML. Terdapat beberapa pilihan format untuk format XML yang akan diekspor. Format ekspor XML yang digunakan ditunjukkan pada Gambar 2.



Gambar 2 Format ekspor XML.

Terdapat kesalahan-kesalahan pada XML yang dihasilkan, yaitu XML yang dihasilkan bukan termasuk XML yang *well-formed* dan terdapat metadata yang memiliki *illegal character* pada XML tersebut.

Dari 7445 dokumen pada XML tersebut terdapat dua dokumen yang tidak *well-formed*, yaitu pada dokumen dengan no registrasi T9603309 dan T2005371. Pada dua dokumen tersebut terdapat element XML yang memiliki *end-tag* yang tidak sesuai. Contoh *element* yang salah adalah sebagai berikut :

```
<Tag_80><![CDATA[656.1/5</REC  
ORD>
```

Element XML tersebut seharusnya memiliki *end-tag* yang sesuai seperti berikut:

```
<Tag_80><![CDATA[656.1/5]]></Tag_80>
```

Sehingga *element* tersebut harus diperbaiki dengan memberikan *end-tag* yang sesuai.

Selain itu terdapat satu dokumen dengan no registrasi T20051187 yang memiliki karakter *illegal* menurut spesifikasi XML, yaitu karakter dengan nilai heksadesimal 0x13. Solusi untuk masalah ini adalah dengan menghilangkan atau mengganti karakter 0x13 tersebut, pada penelitian ini karakter tersebut diganti dengan karakter “_”.

3.2. Pembersihan Data

Dari 7445 dokumen terdapat 28 dokumen (0,37 %) yang tidak memiliki *field* nomor registrasi atau memiliki *field* nomor registrasi tapi isinya kosong. Karena suatu dokumen harus memiliki no registrasi yang unik maka pada metadata tersebut diberikan no registrasi dengan format x1, x2, x3, dan seterusnya.

Selain itu, ada beberapa dokumen yang memiliki nomor registrasi yang sama. Suatu dokumen harus memiliki nomor registrasi yang unik, sehingga ketika ada suatu dokumen yang ternyata nomor registrasi dokumen tersebut sudah ada yang memiliki, maka no registrasi dokumen ini akan diubah menjadi format no_registrasi_awal[a], no_registrasi_awal[b], dan seterusnya. Contoh jika terdapat empat dokumen dengan nomor registrasi 12345 maka ada tiga dokumen yang diubah nomor registrasinya menjadi 12345[a], 12345[b] dan 12345[c]. Terdapat 196 dokumen (2,63 %) yang diubah nomor registrasinya karena memiliki nomor registrasi yang sama.

Hasil dari pembersihan data ini disimpan ke dalam *file* (docs.txt) untuk digunakan sebagai detail dokumen ketika hasil pencarian dokumen sudah didapatkan. Satu baris pada *file* ini menunjukkan detail dari satu dokumen.

Contoh dari data yang disimpan ke dalam *file* adalah sebagai berikut :

```
10300830###Tag_695#Hibiscussabdari  
ffa#Fertilizers#Seedlingvigor##Tag_985  
#1##Tag_850#P##
```

Angka pada awal baris adalah nomor registrasi dari dokumen. Tanda “###” memisahkan *field*, tanda “#” memisahkan nilai suatu *field* jika nilai *field* lebih dari satu. Contoh di atas memiliki arti dokumen tersebut memiliki tiga *field*, yaitu Tag_695, Tag_985, dan Tag_850. *Field* Tag_695 pada dokumen tersebut memiliki tiga nilai, yaitu Hibiscussabdariffa, Fertilizer, dan Seedlingvigor.

3.3. Menghitung Peluang dengan Rumus

Penghitungan peluang dengan rumus diawali dengan melakukan *parsing* pada *field* yang mengandung istilah. Nama *field* istilah yang dipakai adalah *field* subjek atau tag 650 pada metadata INDOMARC.

Proses *parsing* dilakukan untuk mendapatkan istilah-istilah yang ada pada subjek di seluruh dokumen. Pada penelitian ini, istilah adalah kata atau token yang dipisahkan oleh tanda “/ \ [] { } () < > , - ^ ~ ` ; ! ? @ # % & * + = | ”.

Dokumen yang digunakan untuk *indexing* berjumlah 7445. Dari proses *parsing* pada 7445 dokumen ini, didapatkan 7416 dokumen yang memiliki tag 650. Dari 7416 dokumen tersebut didapatkan 4496 istilah unik. Istilah-istilah unik tersebut ditulis ke dalam *file* daftar istilah (istilah.txt) untuk digunakan pada penghitungan bobot dan pencarian dokumen. Penulisan istilah-istilah tersebut dipisahkan dengan titik koma, contoh :

```
acacia;academic;aids;...;zygomycetes;
```

Untuk mempermudah operasi yang menggunakan daftar istilah ini bisa dibuat indeks istilah, istilah “acacia” memiliki indeks istilah 0, “academic” memiliki indeks istilah 1, “aids” memiliki indeks istilah 2 dan seterusnya.

Setelah semua istilah didapatkan, dihitung peluang bersyarat antara istilah-istilah tersebut menggunakan rumus (1). Hasilnya berupa *file* berisi peluang bersyarat (*peluang.txt*) dari 4496 istilah unik yang didapat sebelumnya. Dari hasil penghitungan peluang didapatkan banyak peluang yang bernilai 0, oleh karena itu pembuatan *file* peluang bersyarat istilah hanya menulis nama istilah dan peluang yang nilainya lebih besar dari 0. Format penulisan peluang istilah ini adalah angka sebelum titik koma menunjukkan indeks suatu istilah, sedangkan tanda setelah titik koma menunjukkan nilai peluang bersyarat istilah. Tanda # untuk memisahkan suatu nilai bersyarat suatu istilah dengan istilah lain. Contoh bentuk penulisan peluang istilah pada *file* adalah sebagai berikut :

142;1.0#275;0.2#305;0.25#

Jika baris itu adalah baris milik istilah dengan indeks istilah 0, maka arti dari baris itu adalah nilai peluang bersyarat indeks istilah 0 jika diketahui indeks istilah 142 adalah 1.0. Nilai peluang bersyarat indeks istilah 0 jika diketahui indeks istilah 275 adalah 0,2. Nilai peluang bersyarat indeks istilah 0 jika diketahui indeks istilah 305 adalah 0,25.

3.4. Pembuatan *Inverted Index*

Pembuatan *inverted index* dilakukan terhadap 4496 istilah unik yang didapat ketika *parsing XML*. *Inverted index* berisi istilah unik tersebut beserta nomor registrasi dokumen-dokumen yang memiliki istilah tersebut sebagai isi dari *field* subjeknya. Hasil dari proses ini adalah *file* yang berisi *inverted index* (*inverted.txt*). Contoh dari *inverted index* yang sudah ditulis ke dalam *file* adalah sebagai berikut :

additives;10200341;10300804

Baris itu menyatakan bahwa istilah *additives* terdapat pada dokumen dengan nomor registrasi 10200341 dan 10300804.

Pada Tabel 3 ditampilkan total waktu untuk proses *parsing XML*, pembersihan data, pembuatan peluang istilah, daftar

istilah, dan *inverted index* dengan lima kali iterasi.

Tabel 3 Total waktu *indexing*

Iterasi	Waktu (miliseconds)
1	225890,0
2	220125,0
3	243390,0
4	210828,0
5	204078,0
Rata-rata	220862,2

3.5. Perancangan Visualisasi

Perancangan visualisasi dibagi menjadi tiga tahap, yaitu pemilihan bentuk visual dasar, perbaikan bentuk visual, dan penambahan fitur interaksi.

3.6. Pemilihan Bentuk Visual Dasar

Cara paling mudah untuk merepresentasikan relasi antara dua entitas adalah menggambar garis lurus antara representasi dari dua entitas tersebut (Spence 2007).

Oleh karena itu, bentuk visual dasar yang digunakan pada penelitian ini adalah garis-garis yang menghubungkan antaristilah untuk menunjukkan adanya hubungan antaristilah tersebut. Jika dilihat, maka bentuk visual dasar ini menjadi sebuah bentuk graf, atau lebih tepatnya *tree*, di mana kueri dijadikan *root node* di tengah, dan istilah yang berkaitan dijadikan *child node* seperti yang terlihat pada Gambar 3.



Gambar 3 Bentuk visual dasar.

3.7. Perbaikan Bentuk Visual

Setelah ditentukan bentuk visual dasar yang digunakan maka dilakukan perbaikan pada bentuk visual tersebut, yaitu :

1. Ketebalan *edges* disesuaikan dengan bobot antara istilah tersebut.
 2. Ukuran *child node* merepresentasikan bobot istilah pada *node* tersebut. Semakin besar ukuran *node* dibanding *node* yang lain maka semakin besar bobot istilah tersebut atau dengan kata lain istilah tersebut semakin dekat dengan kueri.
 3. Pemilihan warna untuk *nodes* dan *edges*.
 4. Memberikan warna *nodes* dan warna garis yang berbeda pada *root node*.
- Pebbaikan visual ditunjukkan pada Gambar 4.



Gambar 4 Perbaikan bentuk visual.

3.8. Penambahan Fitur Interaksi

Fitur interaksi yang ditambahkan yaitu :

1. *zoom* (pembesaran) & *pan* (pergeseran).
2. Perubahan warna ketika dilakukan *hover* pada suatu *node* dan ketika *node* tersebut diklik.
3. Ketika dilakukan *click* pada *nodes* maka akan melakukan pencarian dokumen berdasarkan istilah yang ada pada *nodes* tersebut.
4. Ketika dilakukan *double click* pada *nodes* maka akan membuat tree baru di mana *nodes* tersebut akan menjadi *root* baru, lalu istilah lain yang berhubungan *nodes* tersebut akan menjadi *child nodes*.
5. Pengguna dapat mengatur berapa banyak *child nodes* yang akan ditampilkan dan berapa *threshold* yang digunakan.

3.9. Pencarian Dokumen

Pencarian dokumen dilakukan menggunakan pencarian boolean pada bagian subjek. Operasi boolean yang tersedia adalah AND dan OR. Jika terdapat dua *term* pada kueri tetapi tidak diberi *operator* maka *operator* yang digunakan adalah AND.

Contoh ketika diberi kueri “fishes” maka akan dicari dokumen-dokumen di mana *field* subjek dokumen tersebut memiliki kata “fishes”. Jika diberikan kueri “fishes zoology” maka akan dicari dokumen-dokumen di mana subjek pada suatu dokumen tersebut memiliki kata “fishes” dan “zoology”, hasil kueri ini sama saja dengan “fishes AND zoology”. Jika diberikan kueri “fishes OR zoology” maka akan dicari dokumen-dokumen di mana subjek pada dokumen tersebut memiliki kata “fishes” atau “zoology” atau kedua-duanya.

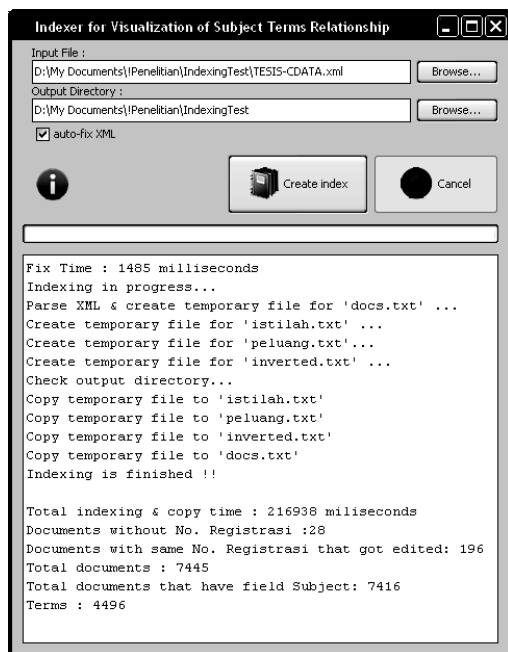
3.10. Penghitungan Bobot Istilah

Penghitungan bobot dilakukan kepada semua istilah unik yang terdapat pada daftar istilah dengan menggunakan rumus (2). Setelah semua istilah di dalam daftar istilah dihitung bobotnya, didapatkan sejumlah istilah dengan bobot tertinggi. Istilah-istilah yang didapatkan tersebut akan divisualisasikan.

3.11. Antarmuka

Aplikasi dibagi menjadi dua, yaitu aplikasi *back-end* untuk melakukan proses *indexing* dan aplikasi *front-end* untuk pencarian dokumen serta penghitungan bobot istilah.

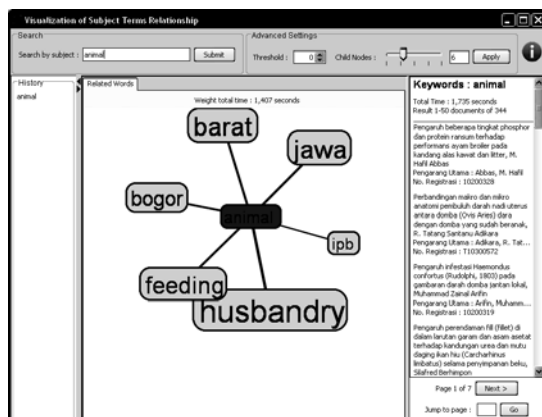
Antarmuka untuk aplikasi *back-end* dapat dilihat pada Gambar 5. Terdapat dua *text field* yang digunakan sebagai masukkan XML *file* dan lokasi hasil *indexing*. Selain itu, terdapat info untuk hasil *indexing*.



Gambar 5 Antarmuka aplikasi *back-end*.

Antarmuka aplikasi *front-end* dibagi menjadi dua bagian, atas dan bawah. Bagian atas digunakan untuk menerima *input* kueri dari pengguna, untuk pengaturan jumlah *child nodes* yang akan ditampilkan, dan untuk pengaturan nilai terkecil (*threshold*) bobot yang akan ditampilkan.

Bagian bawah dibagi menjadi tiga, bagian kiri, tengah, dan kanan. Bagian kiri menampilkan *history* kueri penghitungan bobot, bagian tengah menampilkan visualisasi dari istilah-istilah yang memiliki hubungan dengan kueri sedangkan bagian kanan menampilkan hasil pencarian dokumen. Antarmuka aplikasi dapat dilihat pada Gambar 6.



Gambar 6 Antarmuka aplikasi *front-end*.

Ketika pengguna memasukkan kueri maka akan dilakukan pencarian dokumen dan penghitungan bobot. Hasil dari pencarian dokumen akan ditampilkan di bagian kanan. Hasil pencarian dokumen ini menampilkan 50 dokumen tiap halaman. Dari suatu halaman, pengguna bisa pindah ke halaman berikut, sebelum atau loncat ke halaman tertentu. Untuk menampilkan detail dari dokumen, pengguna dapat menekan judul dokumen yang akan membuat *tab* baru di bagian kiri. *Tab* ini berisi detail metadata dari dokumen tersebut.

Hasil visualisasi bobot istilah akan ditampilkan di bagian tengah. Bentuk visualisasi dan fitur interaksi yang digunakan sesuai dengan yang dijelaskan pada perancangan visualisasi sebelumnya. Kueri-kueri yang sudah digunakan pada penghitungan bobot istilah ditampilkan pada daftar kueri di bagian kiri.

Dilakukan uji coba pada 5 kueri yang terdapat pada daftar istilah untuk mengetahui kecepatan mulai dari penghitungan sampai menampilkan hasil ke antarmuka. Untuk waktu total pencarian dokumen ditampilkan pada Tabel 4, sedangkan untuk waktu total penghitungan bobot sampai divisualisasikan ditampilkan pada Tabel 5.

Tabel 4 Total waktu pencarian dokumen

Istilah kueri	Waktu (<i>milliseconds</i>)
<i>acacia</i>	750,0
<i>zoology</i>	1312,0
<i>animal zoology</i>	766,0
<i>acacia OR zoology</i>	1109,0
<i>industry</i>	1093,0
Rata-rata	1006,0

Tabel 5 Total waktu penghitungan bobot

Istilah kueri	Waktu (<i>milliseconds</i>)
<i>acacia</i>	1156,0
<i>zoology</i>	843,0
<i>animal zoology</i>	734,0
<i>acacia</i> OR <i>zoology</i>	718,0
<i>industry</i>	781,0
Rata-rata	846,4

4. Kesimpulan dan Saran

4.1. Kesimpulan

Dari 7445 dokumen metadata tesis perpustakaan pusat IPB tahun 2006 didapatkan 7416 dokumen yang memiliki *field* subjek atau *tag* 650. Dari 7416 dokumen tersebut, didapatkan 4496 istilah unik pada *field* subjek. Seluruh istilah unik tersebut digunakan pada pembuatan peluang istilah dan *inverted index*.

Hasil implementasi berupa aplikasi *desktop* yang dikembangkan dengan menggunakan bahasa Java. Ketika pengguna memasukkan kueri pada aplikasi maka akan ditampilkan hasil pencarian dokumen yang sesuai dan juga akan ditampilkan visualisasi dari istilah-istilah lain yang berhubungan dengan kueri.

Dari uji coba, didapatkan rata-rata waktu untuk menampilkan hasil pencarian dan menampilkan hasil penghitungan bobot yang sudah divisualisasikan adalah 1 detik. Dengan hasil ini, pengguna tidak perlu menunggu terlalu lama untuk mendapatkan hasil pencarian.

Dengan menggunakan aplikasi ini, diharapkan pengguna dapat mencari dokumen berdasarkan subjek dan juga mengembangkan pencarian yang dilakukannya dengan cara menggunakan istilah-istilah lain yang berhubungan dengan kueri sebagai saran untuk pemilihan kueri selanjutnya.

4.2. Saran

Pada penelitian ini, keterhubungan antaristilah hanya dicari berdasarkan subjek, sehingga pencarian keterhubungan antaristilah ini bisa diperluas ke *field* metadata lainnya seperti *field* kata kunci.

Selain itu, pada tahap *indexing* bisa disempurnakan dengan membedakan istilah yang berupa frase.

Daftar Pustaka

- Adisantoso, J (1997) Temu Kembali Informasi Menggunakan Peluang Bersyarat [tesis]. Jakarta: Program Studi Ilmu Komputer Universitas Indonesia.
- Harold ER, Means WS (2004) *XML in Nuts&Shell*. Sebastopol: O'Reilly Media, Inc.
- Manning CD, Raghavan P, Schutze H. (2008) *Introduction to Information Retrieval*. New York: Cambridge University Press.
- [NISO] National Information Standards Organization (2004) *Understanding Metadata*. Bethesda: NISO Press.
- [PNRI] Perpustakaan Nasional Republik Indonesia (2005) *INDOMARC Format MARC Indonesia*. Jakarta: Perpustakaan Nasional Republik Indonesia.
- Spence R (2007) *Information Visualization Design for Interaction*. Ed ke-2. England: Pearson Education Limited.
- [UNESCO] United Nations Educational, Scientific dan Cultural Organization (2004) *CDS/ISIS for Windows Reference Manual (Version1.5)*. UNESCO.
- Zhang J (2008) *Visualization for Information Retrieval*. Berlin Heidelberg: Springer-Verlag.